Hole In One

DiPS CodeJam 24

Prompt

In the game of Golf, you are given a 2D grid representing the golf course. Each cell of the grid can have one of the following values:
0: An empty cell.
1: A cell with a golf hole.
2: A cell with an obstacle.

You start at a given cell on the grid and need to determine if it's possible to reach any golf hole with a single swing of the golf club. A swing is defined as moving from the starting cell in a straight line (either horizontally, vertically, or diagonally) until you either hit an obstacle, the edge of the grid, or reach a golf hole.
Given an m by n grid and a starting position, can you see if a hole-in-one is possible?

Input Format

- The first line of the input contains 4 space separated integers m n x y, denoting an m by n grid and a starting point of (x, y) such that `grid[x][y]` is possible.
- The next m lines contain n space separated integers denoting one row of the grid.

Output Format

The first and only line of your output must contain a single integer h, 1 if hole-in-one is possible and 0 if not.

Constraints

- $10 \leq m, n \leq 100$

Sample Input/Output

Input	Output

Sample Program

```
def solve(grid, start):
  m, n = len(grid), len(grid[0])
  start_x, start_y = start

  # Directions for moving in straight lines: horizontal, vertical, and diagonal
  directions = [
    (1, 0), (-1, 0),  # Down, Up
```

```
  (0, 1), (0, -1),  # Right, Left
  (1, 1), (-1, -1), # Down-Right, Up-Left
  (1, -1), (-1, 1)  # Down-Left, Up-Right
]

def can_reach_hole(x, y, dx, dy):
  while 0 <= x < m and 0 <= y < n:
    if grid[x][y] == 1:  # Hole found
      return True
    if grid[x][y] == 2:  # Obstacle found
      return False
    x += dx
    y += dy
  return False  # Reached the edge of the grid without finding a hole

# Try all directions from the starting position
for dx, dy in directions:
  if can_reach_hole(start_x, start_y, dx, dy):
    return True

return False
```